Discrete Applied Mathematics 159 (2011) 401–409

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Combinatorics on update digraphs in Boolean networks

J. Aracena [a,*], E. Fanchon [c], M. Montalva [a], M. Noual [b]

[a] CI^2MA and Departamento de Ingeniería Matemática, Universidad de Concepción, Av. Esteban Iturra s/n, Casilla 160-C, Concepción, Chile
[b] Université de Lyon, ÉNS-Lyon, LIP, CNRS UMR5668, and IXXI, Institut rhône-alpin des systèmes complexes, 69007 Lyon, France
[c] TIMC-IMAG, Université Joseph Fourier CNRS-UMR 5525, Domaine de la Merci, 38710 La Tronche, France

ARTICLE INFO

Article history:
Received 30 March 2010
Accepted 20 October 2010
Available online 27 November 2010

Keywords:
Boolean network
Update schedule
Update digraph
Feedback arc set

ABSTRACT

Boolean networks have been used as models of gene regulation and other biological networks. One key element in these models is the update schedule, which indicates the order in which states have to be updated. In Aracena et al. (2009) [1], the authors define equivalence classes that relate deterministic update schedules that yield the same update digraph and thus the same dynamical behavior of the network. In this paper we study algorithmical and combinatorial aspects of update digraphs. We show a polynomial characterization of these digraphs, which enables us to characterize the corresponding equivalence classes. We prove that the update digraphs are exactly the projections, on the respective subgraphs, of a complete update digraph with the same number of vertices. Finally, the exact number of complete update digraphs is determined, which provides upper and lower bounds on the number of equivalence classes.

1. Introduction

Boolean networks (BN) are the simplest models for genetic regulatory networks, as well as for other simple distributed dynamical systems. Despite their simplicity, they provide a realistic model in which different phenomena can be reproduced and studied, and indeed, many regulatory models published in the biological literature fit within this framework [6,9,8].

A BN is defined by its connection digraph, its local activation functions, and the type of update schedule used, which may range from the parallel update, the most common [6,10], to the sequential update, passing through all the combinations of block-sequential updates (which are sequential over the sets of a partition, but parallel inside of each set).

The impact of perturbations of the update schedule on a network dynamics against perturbations in the update schedule has been studied a great deal, mainly from a statistical point of view, in random Boolean networks (RBN), where the local activation functions are probabilistically chosen [2].

Some analytical works on perturbations of update schedules have been made in a particular class of discrete dynamical networks, called sequential dynamical systems, where the connection digraph is symmetric or equivalently an undirected graph and the update schedule is sequential. For this class of networks, the team of Hansson, Mortveit and Reidys studied the set of sequential update schedules preserving the whole dynamical behavior of the network (2001) [7], and the set of attractors in a certain class of cellular automata [5].

* Corresponding author. Fax: +56 412522055.
E-mail addresses: jaracena@ing-mat.udec.cl, jaracena@dim.uchile.cl (J. Aracena), Eric.Fanchon@imag.fr (E. Fanchon), mmontalv@udec.cl (M. Montalva), Mathilde.Noual@ens-lyon.fr (M. Noual).

0166-218X/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2010.10.010

In [1] the authors defined equivalence classes of deterministic update schedules of a BN according to the labeled digraph associated with the network (update digraph). It was proven that two update schedules in the same class yield exactly the same dynamical behavior.

In this paper we focus on the update digraphs and the number and sizes of equivalence classes of update schedules associated with a BN.

The main reason for our interest in update digraphs is twofold. On one hand, we wish to build a better understanding of the objects we are dealing with. On the other hand, we are interested in the relationships that exist between the architecture of the connection digraph of a discrete network and the robustness of its dynamics through the study of the equivalence classes of deterministic update schedules defined by the associated updated digraphs.

2. Definitions

A *digraph* is an ordered pair of sets $G = (V, A)$ where $V = \{1, \dots, n\}$ is a set of elements called *vertices* (or *nodes*) and A is a set of ordered pairs (called *arcs*) of vertices of V. The vertex set of G is referred to as $V(G)$, its arc set as $A(G)$.

A *walk* from a vertex v_1 to a vertex v_m in a digraph G is a sequence of vertices $v_1, v_2, \dots, v_m$ of $V(G)$ such that $\forall k = 1, \dots, m-1$, $(v_k, v_{k+1}) \in A(G)$ or $(v_{k+1}, v_k) \in A(G)$. The vertices v_1 and v_m are the initial and terminal vertices of the walk. A walk is *elementary* if each vertex in the walk appears only once with the possible exception that the first and last vertices may coincide. A walk is closed if its initial and terminal vertices coincide. A *circuit* is a closed elementary walk. A walk $v_1, v_2, \dots, v_m$ is a *path* if $(v_k, v_{k+1}) \in A(G)$ for all $k = 1, \dots, m-1$. A *cycle* is a directed circuit, that is a closed elementary path.

A digraph G is said to be *connected* if there is a walk between every pair of its vertices, and *strongly connected* if there is a path between every pair of its vertices.

$G = (V, A)$ being a digraph and $i \in V$ one of its vertices, $N(i) = \{j \in V \mid (j, i) \in A\}$ denotes the input neighborhood of i in G. More terminology concerning the digraph can be found in [12].

Also, in the sequel, we will write $[\![a, b]\!] = \{a, \dots, b\}$ and $[\![a, b[\![= \{a, \dots, b-1\}$, for any integers a and b.

Definition 1. An *update schedule* of the vertices of a digraph $G = (V, A)$, with $|V| = n$, is a function $s : [\![1, n]\!] \to [\![1, n]\!]$ such that $s(V) = [\![1, m]\!]$ for some $m \le n$. If $\forall i \in V$, $s(i) = 1$, the update schedule is said to be *parallel*. In this case, we will write $s = s_p$. If s is a permutation over the set $\{1, \dots, n\}$, s is said to be *sequential*. And in all other cases, s is said to be *block sequential*.

As mentioned in [3], the number of update schedules associated with a digraph of n vertices is equal to the number of ordered partitions of a set of size n, that is

$$T_n = \sum_{k=0}^{n-1} \binom{n}{k} T_k.$$

Let $G = (V, A)$ be a digraph and s an update schedule. We define $s^{-1}(r) = \{i \in V \mid s(i) = r\}$.

Definition 2. Let $G = (V, A)$ be a digraph and s an update schedule. We define the label function $\mathrm{lab}_s : A \to \{\ominus, \oplus\}$ in the following way:

$$\forall (j, i) \in A, \quad \mathrm{lab}_s(j, i) = \begin{cases} \oplus & \text{if } s(j) \ge s(i) \\ \ominus & \text{if } s(j) < s(i). \end{cases}$$

An arc $a \in A$ such that $\mathrm{lab}_s(a) = \oplus$ is called a *positive arc* and an arc $a \in A$ such that $\mathrm{lab}_s(a) = \ominus$ is called a *negative arc*. Labeling every arc a of A by $\mathrm{lab}_s(a)$, we obtain a labeled digraph (G, lab_s) named the *update digraph* (Fig. 1).

Definition 3. A *Boolean network* $N = (G, F, s)$ is defined by:

- A digraph $G = (V, A)$ with n vertices.
- A global activation function $F = (f_1, \dots, f_n) : \{0, 1\}^n \to \{0, 1\}^n$, where the component functions $f_i : \{0, 1\}^n \to \{0, 1\}$ are called local activation functions and satisfy the following property:
  $j \in N(i) \iff \exists\, (x_1, \dots, x_n) \in \{0, 1\}^n$, such that
  $$f_i(x_1, \dots, x_{j-1}, 0, x_{j+1}, \dots, x_n) \neq f_i(x_1, \dots, x_{j-1}, 1, x_{j+1}, \dots, x_n).$$
- An update schedule $s : V \to [\![1, n]\!]$ of the vertices of G.

The iteration of the discrete network with an update schedule s is given by

$$x_i^{r+1} = f_i(x_1^{l_1}, \dots, x_j^{l_j}, \dots, x_n^{l_n}), \tag{1}$$

where $l_j = r$ if $s(i) \le s(j)$ and $l_j = r+1$ if $s(i) > s(j)$. The exponent r represents the time step.

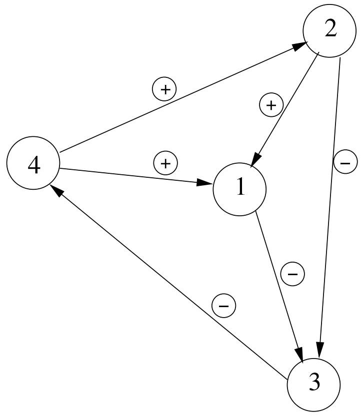

Fig. 1. A digraph $G = (V, A)$ labeled by the function lab_s where $\forall i \in V = \{1, \ldots, 4\}$, $s(i) = i$.

This is equivalent to applying a function $F^s : \{0, 1\}^n \to \{0, 1\}^n$ in a parallel way, with $F^s(x) = (f_1^s(x), \ldots, f_n^s(x))$ defined by

$$f_i^s(x) = f_i(g_{i,1}^s(x), \ldots, g_{i,n}^s(x)),$$

where the function $g_{i,j}^s$ is defined by $g_{i,j}^s(x) = x_j$ if $s(i) \le s(j)$ and $g_{i,j}^s(x) = f_j^s(x)$ if $s(i) > s(j)$. Thus, the function F^s corresponds to the dynamical behavior of the network N. We will say that two networks $N_1 = (G, F, s_1)$ and $N_2 = (G, F, s_2)$ have the same dynamics if $F^{s_1} = F^{s_2}$.

3. Preliminary results and motivations

The following result, given in [1] for Boolean networks, holds:

Theorem 4. *Let $N_1 = (G, F, s_1)$ and $N_2 = (G, F, s_2)$ be two Boolean networks that differ only in the update schedule. If $(G, \mathrm{lab}_{s_1}) = (G, \mathrm{lab}_{s_2})$, then N_1 and N_2 have the same dynamics.*

We define *equivalence classes with respect to labeled digraphs*: if s is an update schedule of the vertices of a digraph G, we write as $[s]_G$ the set of update schedules s' such that $s \overset{G}{\sim} s'$, that is

$$[s]_G = \{s' : (G, \mathrm{lab}_s) = (G, \mathrm{lab}_{s'})\}.$$

An equivalence class, $[s]_G$, is a set of update schedules that all yield the same labeled digraph, and consequently by Theorem 4, the same dynamics on networks.

In this work we study update digraphs and the equivalence classes of their update schedules. More precisely, Section 4 deals with the characterization of update digraphs. Sections 5 and 6 focus on the size and the number of equivalence classes of update schedules.

4. Characterization of update digraphs

In this section, we study the relation $\overset{G}{\sim}$ and the labelings of a given digraph G. First, we give a characterization of the label functions $\mathrm{lab} : A(G) \to \{\oplus, \ominus\}$ that do indeed correspond to label functions induced by update schedules. Then, we examine update schedules s which satisfy $\mathrm{lab} = \mathrm{lab}_s$. The section ends with some observations that were made to help determine the number of $[\cdot]_G$ classes. First, let us give some additional definitions.

Definition 5. A labeled digraph (G, lab) is said to be an *update digraph* (UD) if there exists an update schedule s such that $\mathrm{lab} = \mathrm{lab}_s$, that is $\forall a \in A(G)$, $\mathrm{lab}(a) = \mathrm{lab}_s(a)$ (see the example in Fig. 2).

The goal of this section is to determine the subset of labeled digraphs which are update digraphs.

Definition 6. Let (G, lab) be a labeled digraph and G' a subdigraph of G. We define the projection of (G, lab) onto G' as being the labeled digraph $(G', \mathrm{lab}_{G'})$, where $\mathrm{lab}_{G'}(a) = \mathrm{lab}(a)$, $\forall a \in A(G')$.

Definition 7. Let (G, lab) be a labeled digraph and G' be a non-trivial strongly connected subdigraph of G. The projected labeled digraph (G', lab') is said to be a positive strongly connected component of (G, lab) if $\forall a \in A(G')$, $\mathrm{lab}_{G'}(a) = \oplus$ and it is maximal for this property. We will say that (G, lab) is reduced if it has no positive strongly connected components.

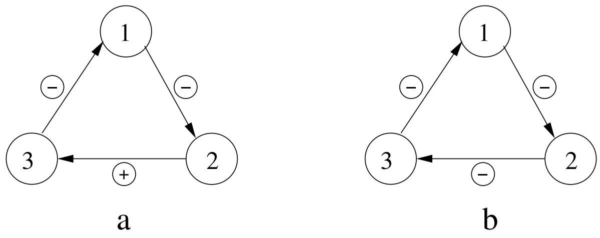

Fig. 2. (a) A labeled digraph (G, lab) which is an update digraph. (b) A labeled digraph (G, lab') which is not an update digraph.

Note that the fact that (G, lab) is an update digraph is independent of the presence or absence of positive strongly connected components, because the images $s(i)$ of the vertices i under an update schedule in a positive strongly connected component are equal. For our study, they can thus be replaced by one unique vertex. In the sequel and without loss of generality, we will work only with reduced labeled digraphs.

Definition 8. Let (G, lab) be a labeled digraph. We define the *labeled reoriented digraph* associated with (G, lab), and write (G_R, lab_R) to refer to the labeled digraph in which all negative arcs are inverted:

- $V(G_R) = V(G)$.
- $A(G_R) = \{(u, v) | (u, v) \in A(G) \wedge \text{lab}(u, v) = \oplus\} \cup \{(u, v) | (v, u) \in A(G) \wedge \text{lab}(v, u) = \ominus\}$.
- $\forall (u, v) \in A(G_R),\ \text{lab}_R(u, v) = \begin{cases} \ominus & \text{if } (v, u) \in A(G) \wedge \text{lab}(v, u) = \ominus, \\ \oplus & \text{otherwise.} \end{cases}$

A forbidden cycle in (G_R, lab_R) is a cycle containing a negative arc.

An example of a labeled reoriented digraph is shown in Fig. 3.

Let (G, lab) be a labeled digraph. We can determine whether it is reduced in time $\mathcal{O}(|A|)$ with an algorithm that searches for strongly connected components of a digraph. We can also get (G_R, lab_R) in time $\mathcal{O}(|A|)$.

Definition 9. Let (G, lab) be a labeled digraph and P a path in (G_R, lab_R). We denote by $l^-(P)$ the number of negative arcs of P. Thus, for every $v \in V(G)$ we define the set $\mathcal{P}_v$ of paths ending in v, and $L^-(v) = \max_{P_v \in \mathcal{P}_v} l^-(P_v)$ and

$$L^-(G_R, \text{lab}_R) = \max_{v \in V(G)} \{L^-(v)\},$$

the number of negative arcs of a path with the maximum number of negative arcs over all paths in (G_R, lab_R).

Theorem 10. *A labeled digraph (G, lab) is an update digraph if and only if (G_R, lab_R) does not contain any forbidden cycle.*

Proof. $(\Rightarrow)$ Let us suppose that (G_R, lab_R) contains a forbidden cycle $C : v_1, \dots, v_p = v_1$ such that (v_j, v_{j+1}) is a negative arc. Then any update schedule s such that $(G, \text{lab}) = (G, \text{lab}_s)$ must satisfy $s(v_j) > s(v_{j+1})$. It must also satisfy $s(v_j) \leq s(v_{j+1})$ since there exists in (G_R, lab_R) a path from v_{j+1} to v_j. Thus, we end up with a contradiction.

$(\Leftarrow)$ Let $L = L^-(G_R, \text{lab}_R)$. Observe first that if $P : v_1, \dots, v_k$ is a path in G_R such that $l^-(P) = L$ with $\{(v_{i_1}, v_{i_2}), (v_{i_3}, v_{i_4}), \dots, (v_{i_{2L-1}}, v_{i_{2L}})\}$ the set of negative arcs of P where $j > k \Rightarrow i_j > i_k$, and s is an update schedule such that $(G, \text{lab}) = (G, \text{lab}_s)$, then

$$s(v_{i_1}) > s(v_{i_2}) > s(v_{i_4}) > s(v_{i_6}) > \cdots > s(v_{i_{2L}}),$$

which implies $\max\{s(v) | v \in V(G)\} \geq L + 1$. Besides,

$$\forall i = 1, \dots, k, \quad L^-(v_i) = l^-(v_1, \dots, v_i) \quad \text{and} \quad L^-(v_1) = 0.$$

Let $s : V(G) \to [\![1, L+1]\!]$ with

$$s(v) = L - L^-(v) + 1, \quad \forall v \in V(G).$$

We observed above that $s(V(G)) = [\![1, L+1]\!]$, meaning that s is an update schedule of $V(G)$. To check that s is also an update schedule satisfying $(G, \text{lab}) = (G, \text{lab}_s)$, we must show that $\forall a = (u, v) \in A(G_R)$, $s(u) > s(v) \Leftrightarrow \text{lab}_G(u, v) = \ominus$. This follows from the fact that, (u, v) being an arc of G_R, it necessarily holds that $L^-(v) \geq 1 + L^-(u)$ when $\text{lab}_G(u, v) = \ominus$. □

We notice that if (G, lab) is a labeled digraph, the forbidden cycles of (G_R, lab_R) correspond to what we will refer to as alternating circuits of G. That is, they coincide with walks of G, $C = v_0, v_1, \dots, v_k$, where $v_0 = v_k$ and either $(v_i, v_{i+1}) \in A$ in which case $\text{lab}_G(v_i, v_{i+1}) = \oplus$ or $(v_{i+1}, v_i) \in A$ in which case $\text{lab}_G(v_{i+1}, v_i) = \ominus$ (or vice versa). Among these alternating circuits are, in particular, circuits such that $\forall i \in [\![0, k-1]\!]$, $\text{lab}(v_i, v_{i+1}) = \ominus$ as well as subgraphs containing two vertices u and v, a walk from u to v negatively labeled and another walk from u to v positively signed.

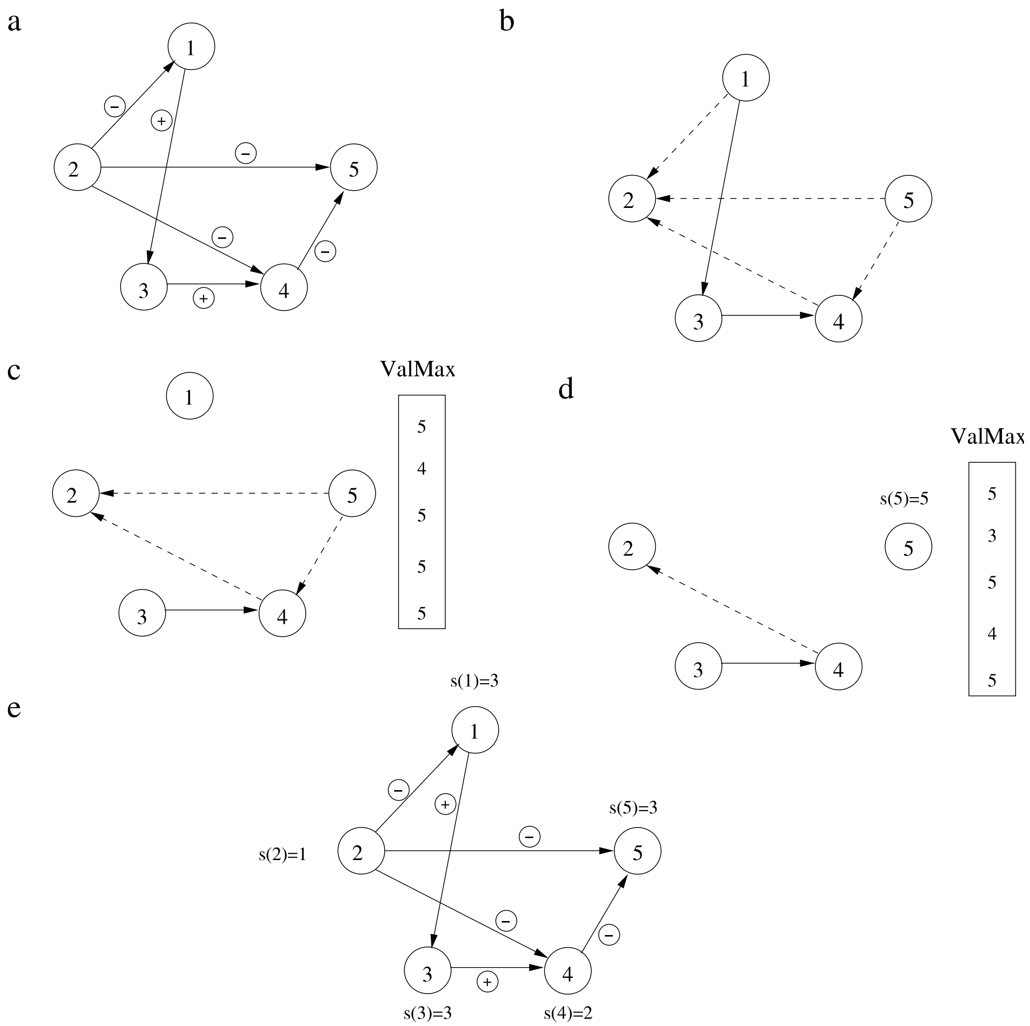

Fig. 3. (a) A labeled digraph $G = (\{1, \ldots, 5\}, A)$. (b) (G_R, lab_R). The arcs drawn in dotted lines are negative arcs. The others are positive arcs. (c) and (d) show the first two steps computed by Algorithm 1 after the *while loop*. (e) The update schedule s such that $(G, \mathrm{lab}) = (G, \mathrm{lab}_s)$.

Incidentally, let us notice that, as a consequence of Theorem 10, if $a = (u, v) \in A(G)$ is an arc not belonging to any circuit, then whether (G, lab) is an update digraph or not is independent of $\mathrm{lab}(a)$.

Algorithm 1, given below, finds an update schedule corresponding to a given reduced labeled digraph as described in the proof of Theorem 10. It is adapted from the famous algorithm [11], giving a topological order on a digraph without cycles. For a given reduced labeled digraph (G, lab), Algorithm 1 works on the labeled reoriented digraph (G_R, lab_R) without forbidden cycles. It returns in time $\mathcal{O}(|V| + |A|)$ an update schedule s such that $(G, \mathrm{lab}) = (G, \mathrm{lab}_s)$ and

$$\max\{s(v) \mid v \in V\} = \min\{\max\{s'(v) \mid v \in V\} \mid s' \text{ is an update schedule of } G\}.$$

Fig. 3 shows the different steps of the algorithm that returns an update schedule associated with an arbitrary possible labeled digraph (not necessarily reduced).

Corollary 11. *The following problems can be solved in polynomial time.*

(1) *Determine whether a labeled digraph* (G, lab) *is an update digraph.*
(2) *Given* (G, lab) *an update digraph, find an update schedule s such that* $(G, \mathrm{lab}) = (G, \mathrm{lab}_s)$.

Indeed, according to Theorem 10, a labeled digraph (G, lab) is an update one if and only if in (G_R, lab_R) no negative arc belongs to a strongly connected component. Thus, the first part of Corollary 11 holds since the strongly connected components of a digraph can be identified in polynomial time. For the second one, an update schedule s such that $(G, \mathrm{lab}) = (G, \mathrm{lab}_s)$ can be constructed by using the Algorithm 1 whose run time is also polynomial.

Algorithm 1. `update schedule associated with a labeled digraph`
Input: $(G = (V, A), lab)$ a reduced labeled digraph such that (G_R, lab_R) has no forbidden cycle
begin

```
    ValMax ← table of size |V(G_R)| in which are stored the maximal possible values of s(v), v ∈ V(G_R).
    n ← |V|;
    H ← G_R;
    forall v ∈ V do
        ValMax[v] = n;
    end

    while ∃v ∈ V, N_H(v) = ∅ do
        s(v) ← ValMax[v];
        forall (v, w) ∈ A(H) do
            if (w, v) ∈ A(G) is a negative arc then
                ValMax[w] ← min{ValMax[w], s(v) − 1; }
            else
                ValMax[w] ← min{ValMax[w], s(v); }
            end
            delete the arc (v, w) from H
        end
    end

    s_min ← min{s(v | v ∈ V); }
    forall v ∈ V do
        s(v) ← s(v) − s_min + 1;
    end
```

end

5. Sizes of the equivalence classes $[\cdot]_G$

Let us now consider the following question: given a digraph G and an update schedule s, does there exist any update schedule $s' \neq s$ such that $(G, \text{lab}_s) = (G, \text{lab}_{s'})$? That is, what conditions need to be satisfied in order for $|[s]_G| > 1$ to hold?

Corollary 12. *Let* (G, lab) *be a reduced update digraph with* $|V(G)| = n$ *and* $L = L^-(G_R, \text{lab}_R)$*. Then,* $\forall m \in [\![L, n-1]\!]$*, there exists an update schedule s such that* $\max\{s(v)|v \in V\} = m + 1$ *and* $(G, \text{lab}) = (G, \text{lab}_s)$*.*

Proof. We show the result by induction on m.
If $m = L$, the result was proved in Theorem 10.

If $L = n - 1$, we are done. Otherwise, let $m \in]\!]L, n-1]\!]$. By the induction hypothesis, there exists an update schedule s such that $(G, \text{lab}) = (G, \text{lab}_s)$ and $\max\{s(v)|v \in V(G)\} = m$. Since $m < n$, there exists $i^* \in [\![1, n-1]\!]$ such that $|s^{-1}(i^*)| > 1$. Notice that $\forall (u, v) \in s^{-1}(i^*) \times s^{-1}(i^*) \cap A(G)$, $\text{lab}_s(u, v) = \oplus$. Besides, because there are no cycles in (G_R, lab_R), there exists $w \in s^{-1}(i^*)$ such that $\{v \in s^{-1}(i^*)|(w, v) \in A(G_R)\} = \emptyset$. Hence, let us define s' as follows:

$$s'(v) = \begin{cases} s(v) + 1 & \text{if } s(v) \geq s(w) \text{ and } v \neq w, \\ s(v) & \text{if } s(v) < s(w) \text{ or } v = w. \end{cases}$$

Then obviously, $s'(V(G)) = [\![1, m+1]\!]$, i.e. s' in an update schedule of $V(G)$, and $(G, \text{lab}) = (G, \text{lab}_{s'})$. $\square$

Corollary 13. *Let* (G, lab) *be a reduced update digraph and* $L = L^-(G_R, \text{lab}_R)$*. Then,* $|[s]_G| \geq |V(G)| - L$*, where s is an update schedule such that* $(G, \text{lab}) = (G, \text{lab}_s)$*.*

Corollary 14. *Let* $(G = (V, A), \text{lab}_s)$ *be a reduced update digraph.* $|[s]_G| > 1$ *if and only if* (G_R, lab_R) *is not a negative linear digraph.*

Proof. If (G_R, lab_R) is not a negative linear digraph, i.e. it has no directed path of length $|V| - 1$ with all its arcs negative, then $L \leq |V| - 2$. Thus, by Corollary 13, $|[s]_G| > 1$, where $(G, \text{lab}) = (G, \text{lab}_s)$.

Conversely, if G is a negative linear digraph with $p : u_1, \ldots, u_{|V|}$ a directed path of length $|V| - 1$ with $\text{lab}_G(u_i, u_{i+1}) = \ominus$, $\forall i = 1, \ldots, |V| - 1$, then there only exists an update schedule s that satisfies $(G, \text{lab}) = (G, \text{lab}_s)$. $\square$

As a consequence, $|[s_p]_G| > 1$ if and only if G is not strongly connected.

6. The number of update digraphs

In the previous section, given a labeled digraph (G, lab), we were interested in the existence of update schedules s such that $(G, \text{lab}) = (G, \text{lab}_s)$. And when such update schedules did exist, we wanted to know how many there were.

In the present section, given a digraph G, we would like to determine how it can be labeled as an update digraph, that is, which are the label functions lab of G such that (G, lab) is indeed an update digraph. In particular, here, we focus on the number of equivalence classes $[\cdot]_G$ (rather than their sizes).

Definition 15. We define the size of a labeled digraph (G, lab) by the number of its positive arcs.

We define the following problem:

DIGRAPH UPDATE (DU) problem:
- **Input:** A digraph $G = (V, A)$ and an integer k;
- **Question:** Does there exist a label function $\text{lab} : A \to \{\oplus, \ominus\}$ such that (G, lab) is an update digraph and its size is at most k?

Theorem 16. *DIGRAPH UPDATE is NP-complete.*

Proof. We are going to prove Theorem 16 by reduction to the FAS problem defined below and which is known to be NP-complete [4]:

FAS problem:
- **Input:** A digraph $G = (V, A)$ and an integer k;
- **Question:** Does there exist a feedback arc set F of G such that $|F| \le k$?

where a feedback arc set (FAS) F of G is a set of arcs such that the digraph $(V, A \setminus F)$ does not have any cycles. F is minimal if there does not exist a FAS F' of G such that $F' \subsetneq F$.

The reduction function that we use to map an instance of FAS to an instance of DU is simply the identity. Next, for a given instance (G, k) we show that there exists a label function lab such that (G, lab) is an update digraph of size at most k if and only if there exists a FAS F of G such that $|F| \le k$.

($\Rightarrow$) Let lab be a label function such that (G, lab) is an update digraph of size at most k and let $F = \{a \in A(G) | \text{lab}(a) = \oplus\}$. Then, F is a FAS of size $|F| \le k$. $G' = (V, A \setminus F)$ cannot contain any cycle since otherwise it would be a negative cycle of (G, lab) which is not possible in an update digraph.

($\Leftarrow$) Let F be a minimal FAS of G such that $|F| \le k$. Let $a \in F$. If every cycle of G containing a contains also another arc of F then $F \setminus \{a\}$ is a FAS of G smaller than F. This contradicts the minimality of F. Thus, for every $a \in F$, there exists a cycle of G containing a and no other arc of F. Now, let us define the label function lab as follows:

$$\forall a \in F, \text{lab}(a) = \oplus \quad \text{and} \quad \forall a \in A \setminus F, \text{lab}(a) = \ominus.$$

Note that because there are no cycles in $G' = (V, A \setminus F)$, there are no negative cycles in (G, lab). Suppose, however, that (G, lab) is not an update digraph. In (G, lab), there must thus be an alternating circuit (see Theorem 10 and the remarks made thereafter) containing both positive and negative arcs. In other words, there is a forbidden cycle in (G_R, lab_R). The positive arcs in this cycle belong to F. Let $a \in A(G)$ be such a positive arc belonging to the forbidden cycle and to F. From the discussion above, we derive that there exists a cycle C_a of G that contains a and no other arc of F. All the arcs of C_a that are not a are thus negative in (G, lab). Concatenating the negative arcs of the alternating circuit and of all cycles C_a (a being an arc of F in the forbidden cycle) we obtain a cycle in $G' = (V, A(G) \setminus F)$ (see Fig. 4) which contradicts F being a FAS of G (as well as the fact that (G, lab) has no negative cycles). □

Corollary 17. *Let G be an update digraph. If N_{FAS} and N_{MFAS} are, respectively, the total number of* FAS *and minimal* FAS *of G, then*

$$N_{\text{MFAS}} \le |\{[s]_G \mid s \text{ is un update schedule of } V(G)\}| \le N_{\text{FAS}}.$$

An example of digraph G where the number of update digraphs is distinct from the number of FAS and minimal FAS is shown in Fig. 5. For this digraph, $N_{\text{MFAS}} = 3$, $N_{\text{FAS}} = 8$ and the number of associated update digraphs is 6.

7. Extensions and projections of update digraphs

Theorem 18. *Let G be a digraph and G' a subdigraph of G. If (G', lab') is an update digraph, then there exists a label function* lab *of $A(G)$ such that (G, lab) is an update digraph and $\text{lab}_{G'} = \text{lab}'$.*

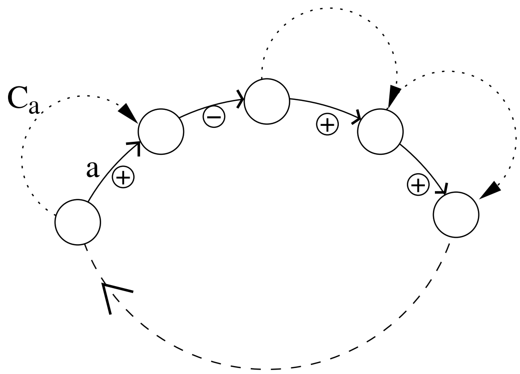

Fig. 4. A forbidden cycle in (G_R, lab_R) with, surrounding it, the negative cycles C_a mentioned in the proof of Theorem 16. Arrows shown with full lines represent arcs, arrows shown with dashed lines represent paths.

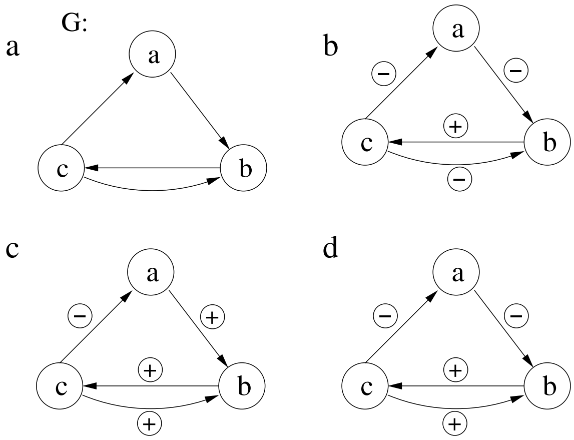

Fig. 5. (a) A digraph G. (b) An update digraph, where $\{(b, c)\}$ is a minimal FAS. (c) An update digraph, where $\{(a, b), (b, c), (c, b)\}$ is a FAS but not minimal. (d) A non-update digraph, but $\{(b, c), (c, b)\}$ is a FAS.

Proof. If (G', lab') is an update digraph we will show that for all $a = (u, v) \in A(G) \setminus A(G')$, either $(G' + a, \mathrm{lab}_a^+)$ or $(G' + a, \mathrm{lab}_a^-)$ is an update digraph, where $V(G' + a) = V(G') \cup \{u, v\}$, $E(G' + a) = E(G') \cup \{a\}$ and lab_a^+ and lab_a^- are defined by $\mathrm{lab}_a^+(e) = \mathrm{lab}_a^-(e) = \mathrm{lab}(e),\ \forall e \in A(G')$, $\mathrm{lab}_a^+(a) = \oplus$ and $\mathrm{lab}_a^-(a) = \ominus$.

Let us suppose that there exists $a = (u, v) \in A(G) \setminus A(G')$ such that neither $(G' + a, \mathrm{lab}_a^+)$ nor $(G' + a, \mathrm{lab}_a^-)$ is an update digraph. Then there exists a forbidden cycle $C_1 : x_1 = u, x_2 = v, x_3, \ldots, x_p = u$ with $\mathrm{lab}_a^+(x_j, x_{j+1}) = \ominus$ in the reoriented labeled digraph $((G' + a)_R, (\mathrm{lab}_a^+)_R)$. In the same way, there exists a forbidden cycle $C_2 : y_1 = v, y_2 = u, y_3, \ldots, y_q = v$ in the reoriented labeled digraph $((G' + a)_R, (\mathrm{lab}_a^-)_R)$. Hence, the sequence of nodes $x_2 = v, \ldots, x_j, x_{j+1}, \ldots, x_p = u = y_2, \ldots, y_q = v$ in the reoriented labeled digraph (G'_R, lab'_R) contains a cycle including the arc (x_j, x_{j+1}) (see Fig. 6), that is a forbidden cycle. Thus (G', lab') is not an update digraph, which is a contradiction.

Therefore, if $A(G) \setminus A(G') = \{a_1, \ldots, a_r\}$, then by induction we can prove that for all k in $\{1, \ldots, r\}$ there exists a label function lab_k of the arcs of $G' + a_1 + \cdots + a_k$ such that $(\mathrm{lab}_k)_{G'} = \mathrm{lab}'$ and $(G' + a_1 + \cdots + a_k, \mathrm{lab}_k)$ is an update digraph. In particular, there exists a label function lab in G such that (G, lab) is an update digraph and $\mathrm{lab}' = \mathrm{lab}_{G'}$. □

Note that if (G, lab) is an update digraph and $\mathrm{lab}' = \mathrm{lab}_{G'}$, then (G', lab') is also an update digraph by Theorem 10. Therefore, the update subdigraphs are the projections of the update digraphs.

Corollary 19. *Let G be a connected digraph of $n > 1$ vertices. Then,*

$$2^{n-1} \leq |\{(G, \mathrm{lab}) : (G, \mathrm{lab}) \textit{ is update digraph }\}| \leq T_n$$

where $T_n = \sum_{k=0}^{n-1} \binom{n}{k} T_k$.

Proof. From Theorem 18 for all digraphs G and subdigraphs G',

$$|\{(G', \mathrm{lab}') : (G', \mathrm{lab}') \text{ is UD}\}| \leq |\{(G, \mathrm{lab}) : (G, \mathrm{lab}) \text{ is UD}\}|.$$

On the other hand, the connected digraph of n vertices with the least number of arcs, i.e. $n - 1$, is an oriented tree. In this case, all labeling functions on the digraph yield an update digraph. Thus, there are 2^{n-1} update connected digraphs with

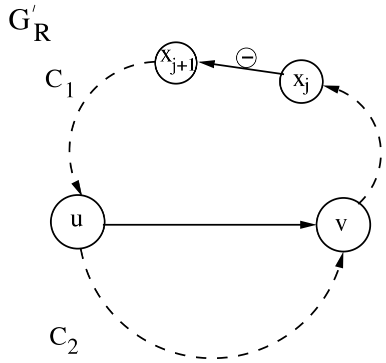

Fig. 6. Scheme of the forbidden cycle in (G'_R, lab'_R) mentioned in the proof of Theorem 18.

the least number of arcs. In the same way, the connected digraph of n nodes with the greatest number of arcs, equal to n^2 (including the arcs (u, u)), is the complete digraph. In this case, each label function on a complete digraph defines a total preorder on the vertices. Besides, it is well known that the number of total preorders on a set of n elements is T_n defined as in the statement of this theorem. Thus, T_n is the maximum number of update connected digraphs with n vertices. □

Acknowledgements

This work was partially supported by FONDECYT project 1090549 (J.A.), by FONDAP and BASAL projects CMM, Universidad de Chile, by Centro de Investigación en Ingeniería Matemática (CI^2MA), Universidad de Concepción, and by a CONICYT fellowship and MECESUP Project UCO0713 (M.M.).

References

[1] J. Aracena, E. Goles, A. Moreira, L. Salinas, On the robustness of update schedules in Boolean networks, BioSystems 97 (2009) 1–8.
[2] M. Chaves, R. Albert, E. Sontag, Robustness and fragility of Boolean models for genetic regulatory networks, Journal of Theoretical Biology 235 (2005) 431–449.
[3] J. Demongeot, A. Elena, S. Sené, Robustness in regulatory networks: a multi-disciplinary approach, Acta Biotheoretica 56 (1–2) (2008) 27–49.
[4] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman, 1979.
[5] A. Hansson, H. Mortveit, C. Reidys, On asynchronous cellular automata, Advances in Complex Systems 8 (4) (2005) 521–538.
[6] S. Kauffman, Metabolic stability and epigenesis in randomly connected nets, Journal of Theoretical Biology 22 (1969) 437–467.
[7] H. Mortveit, C. Reidys, Discrete, sequential dynamical systems, Discrete Mathematics 226 (2001) 281–295.
[8] I. Shmulevich, H. Lähdesmäki, E. Dougherty, J. Astola, W. Zhang, The role of certain post classes in Boolean network models of genetic networks, Proceedings of the National Academy of Sciences of the United States of America 100 (2003) 10734–10739.
[9] R. Thomas, Boolean formalization of genetic control circuits, Journal of Theoretical Biology 42 (1973) 563–585.
[10] R. Thomas, Regulatory networks seen as asynchronous automata: a logical description, Journal of Theoretical Biology 153 (1991) 1–23.
[11] J. van Leeuwen, Handbook of Theoretical Computer Science, Elsevier, 1990.
[12] D.B. West, Introduction to Graph Theory, Prentice Hall, 1996.